## Практические задания для Урока 1 Модуля 2

1. Напишите программу, которая возводит число a в степень n, a и n вводятся с клавиатуры в отдельных строках. Возведение в степень оформите в виде функции. Программа должна вывести одно число=а в степени n.
2. Напишите программу, которая запрашивает у пользователя два массива по 10 элементов в каждом (целые числа вводятся с клавиатуры, каждое число — в новой строке, сначала значения первого массива, затем — второго). Программа попарно сравнивает элементы массивов и формирует третий массив, состоящий из наибольших элементов пар. Вывести на экран значения элементов третьего массива в строку через один пробел между элементами. Алгоритм ввода значений массивов и формирования третьего массива оформить в виде функций.
3. Напишите программу, которая определяет расстояние между двумя точками. Программа должна запрашивать ввод координат точек x1,y1 и x2,y2 (значения координат вводятся с клавиатуры попарно, каждая пара координат в отдельной строке). Программа должна вывести на экран расстояние между точками (в формате 10 знакомест под все число и 2 знакоместа под дробную часть). Расчет расстояния оформите в виде функции.

Для вычисления квадратного корня подключите библиотеку #include <math.h> и используйте функцию sqrt().

1. Пользователь вводит натуральное число n (n>1). Проверьте, является ли оно простым. Программа должна вывести “+”, если число простое и “-”, если число составное. Решение оформите в виде функции, которая возвращает “+” для простых чисел и “-” для составных чисел.
2. Напишите программу, которая находит минимальный элемент в массиве. Программа должна запрашивать ввод значений элементов массива с клавиатуры (10 целочисленных элементов, каждый вводится в отдельной строке). Программа должна вывести значение наименьшего элемента. Поиск наименьшего элемента оформите в виде функции, которая возвращает значение минимального элемента, а в качестве параметра получает массив.

## Решения

1.

#include <stdio.h>

int st(int a,int n)

{

int s=1;

for (int i=0;i<n;i++)

s=s\*a;

return s;

}

int main()

{

int a,n;

printf("a=");

scanf("%d",&a);

printf("n=");

scanf("%d",&n);

printf("%d\n",st(a,n));

return 0;

}

2.

#include <stdio.h>

void input(int m[10])

{

for (int i=0;i<10;i++)

scanf("%d",&m[i]);

}

void check(int m1[10],int m2[10],int m3[10])

{

for (int i=0;i<10;i++)

{

if (m1[i]>m2[i])

m3[i]=m1[i];

else

m3[i]=m2[i];

}

}

int main()

{

int m1[10]={};

int m2[10]={};

int m3[10]={};

input(m1);

input(m2);

check(m1,m2,m3);

for (int i=0;i<10;i++)

printf("%d ",m3[i]);

return 0;

}

3.

#include <stdio.h>

#include <math.h>

float distance(int x1,int y1,int x2,int y2)

{

return (sqrt((x2-x1)\*(x2-x1)+(y2-y1)\*(y2-y1)));

}

int main()

{

int x1,y1,x2,y2;

printf("x1, y1= ");

scanf("%d%d", &x1,&y1);

printf("x1, y1= ");

scanf("%d%d", &x2,&y2);

printf("%10.2f\n",distance(x1,y1,x2,y2));

return 0;

}

4.

#include <stdio.h>

#include <math.h>

char prost(int a)

{

int r=0;

for (int i=2;i<=(sqrt(a)+1)/1;i++)

if (a%i==0)

{

r=1;

break;

}

if (r==0) return '+';

else return '-';

}

int main()

{

int a;

scanf("%d",&a);

printf("%c\n",prost(a));

return 0;

}

5.

#include <stdio.h>

int min(int m[10])

{

int min\_i=m[0];

for (int i=1;i<10;i++)

if (m[i]<min\_i) min\_i=m[i];

return min\_i;

}

int main()

{

int m[10]={};

for (int i=0; i<10;i++)

scanf("%d",&m[i]);

printf("%d\n",min(m));

return 0;

}